Research Statement

Pierre Vial

Development of System S and comparison with System $\mathscr{R}_0$

The main contribution of my PhD is the introduction of an intersection type system that I named System S, because it resorts to *sequences* to represent intersection. Here, a sequence of types is meant as family of types indexed by a family of integers. For instance, sequence type $(T_k)_{k\in\{2,3,8\}}$, with $T_2 = S$, $T_3 = T$ and $T_8 = S$, is an intersection of two occurrences of type S (*placed* on tracks 2 and 8) and one occurrence of type T (placed on track 3). This sequence type is also denoted $(2{\cdot}S, 3{\cdot}T, 8{\cdot}S)$.

This system has a quantitative flavour and is closely related to System $\mathscr{R}_0$, that was independently introduced by Gardner and de Carvalho in 1994 and 2007 respectively. System $\mathscr{R}_0$ uses *multisets* to represent intersection *e.g.* in $\mathscr{R}_0$, the multiset $[\sigma, \tau, \sigma]$ will represent the intersection of two occurrences of the type σ and one occurrence of τ. This multiset is distinct from $[\sigma, \tau]$ and the multiplicity of a type in an intersection matters in System $\mathscr{R}_0$. From the algebraic perspective, the intersection operator of $\mathscr{R}_0$ is *non idempotent*. A bit similarly, the intersection type $(2{\cdot}S, 3{\cdot}T, 8{\cdot}S)$ of System S explicitly contains two occurrences of S.

Let us recall some fundamental properties of System $\mathscr{R}_0$:

- As the other intersection type systems, System $\mathscr{R}_0$ is designed to satisfy both Subject Reduction (if $\Gamma \vdash t : \tau$ and $t \to_\beta t'$, then $\Gamma \vdash t' : \tau$) and Subject Expansion (if $\Gamma \vdash t' : \tau$ and $t \to_\beta t'$, then $\Gamma \vdash t : \tau$). Subject Expansion usually cannot be satisfied in a simply type system, and is one of the main reasons why intersection type systems are able to *characterize* normalizability whereas simply type systems only *ensures* it.

- System $\mathscr{R}_0$ is *relevant* meaning that *weakening* is forbidden. Concretely, this means that, from the perspective of β-reduction, derivations behave really similarly to λ-terms. This explains why Subject Reduction and Subject Expansion are proven very elegantly in System $\mathscr{R}_0$.

- If Π is a $\mathscr{R}_0$-derivation typing a term t and we reduce a redex of t that is *typed* in Π, then we obtain a reduct t' and a derivation Π' typing t' that is strictly smaller than Π. By smaller, we mean that Π contains less rules than Π does. Since the number of rules that a derivation contain is a natural number, we cannot keep on reducing indefinitely typed redexes in a term. This argument may be made more precise and yield many termination properties (*e.g.* a term is head normalizing iff head reduction terminates), using only a simple arithmetical argument. In contrast, in idempotent intersection type system or higher-order intersection type systems, termination properties are usually proven resorting to far more technical realizability argument.

System S naturally features "pointers" *e.g.* the integer 2 and 8 to distinguish the two occurrences of S (hence the vocable "rigidity"). We have a "disjoint union" operator $\uplus$ for sequences *e.g.* $(2{\cdot}S, 3{\cdot}T) \uplus (8{\cdot}S) = (2{\cdot}S, 3{\cdot}T, 8{\cdot}S)$. In contrast, with multisets, we have $[\sigma, \tau] + [\sigma] = [\sigma, \sigma, \tau]$, but, in this equality, we have no way to relate one occurrence of σ in $[\sigma, \sigma, \tau]$ to $[\sigma, \tau]$ rather than $[\sigma]$ and *vice versa*.

Contrary to system $\mathscr{R}_0$, S is coinductive: we allow infinite type and infinite sequences. As it turns out, it is particularly fit to study infinite λ-calculus or non-terminating λ-terms. We name $\mathscr{R}$ the coinductive version of system $\mathscr{R}_0$.

Klop's Problem

Klop's Problem is the following: can we characterize the set of *hereditary head normalizing (HHN) terms* with a type system? Hereditary head normalization is a *coinductive* predicate: a term t is HHN if it reduces to a head normal form $\lambda x_1 \dots x_p.x\, t_1 \dots t_q$ and $t_1, \dots, t_q$ are also HHN. Thus, roughly speaking, hereditary head normalization is the coinductive version of weak normalization. Tatsuta proved in 2007 that the set of HHN terms is not recursively enumerable. Since in an inductive type system, the set of typable terms is recursively enumerable, Tatsuta concluded that an *inductive* type system could not characterize the set of HHN terms. However, System $\mathtt{S}$ features a coinductive type grammar and it turns out that:

Theorem (Vial, LICS17)**.** A term t is hereditary head normalizing iff it is *unforgetfully* typable in System $\mathtt{S}$ by means of an *approximable* derivation.

A derivation is *unforgetful* when it guarantees that every part of a normal form is typed (whereas some derivations may type only its head variable). Unforgetfulness is a criterion well-known for usual, inductive intersection type systems, whereas *approximability* is a *validity criterion* used to discard some *unsound* typing derivation. Indeed, due to the presence of coinductive types, it is easy to type some very unsound terms in System $\mathtt{S}$, as $\Omega = \Delta, \Delta$ where $\Delta = \lambda x.x\, x$. An interesting point is that approximability *cannot* be formulated in System $\mathscr{R}$, the coinductive version of System $\mathscr{R}_0$, but only in System $\mathtt{S}$. We may wonder now if we can characterize infinitary strong normalization.

System $\mathtt{S}$ can also be used to characterize the set of hereditary permutation, which gives a positive answer to TLCA Problem # 20 (Tatsuta proved that the answer to this problem was negative in the inductive setting).

Complete unsoundness or typing the untyped

As noted above, Systems $\mathtt{S}$ or $\mathscr{R}$ can type unsound terms *e.g.* Ω is typable in those systems. It turns out that, if we discard the validity criterion that we named approximability, we have:

Theorem (Vial)**.** Every λ-term is typable in system $\mathscr{R}$ (or in System $\mathtt{S}$). Moreover, if a term t is of *order*, then it is typable with a type variable.

The *order* of a λ-term t is the *supremal* integer n such that $t \to_\beta^* \lambda x_1 \dots x_n.t'$. It is thus a semantical information about t. In a type system enjoying Subject Expansion, a *typable* term that is not of order 0 may only be typed with an arrow type (and not with a type variable), because typing an abstraction usually produces an arrow. Thus, systems $\mathscr{R}$ and $\mathtt{S}$ are able to "capture" the order of *any* λ-term. This work provides a new model for the pure (untyped, so to say) λ-calculus, whose equational theory still needs to be studied.

If a type system is able to type any term, we say that it is *completely unsound*. When a type system is not *relevant* and features coinductive types, then complete unsoundness is usually straightforward and proven by reasoning by induction on the structure of the terms. Systems $\mathscr{R}$ and $\mathtt{S}$ are both relevant (weakening is forbidden) and complete unsoundness turns out to be a difficult problem. Indeed, in order to prove that some set of terms is typable, the usual technique is the following: typing a set of terms whose some parts are stabilized (*i.e.* normalized) and then proceed by subject expansion. Normal forms are usually easy to type because they are made of "blocks" of the form $\lambda x_1 \dots x_p.x\, t_1 \dots t_q$: we assign then to x an arrow type that is "fed" with the types of $t_1, \dots, t_q$.

In contrast, systems $\mathtt{S}$ and $\mathscr{R}$ do not ensure any form of "stabilization" and this usually technique cannot be applied. One of the main contribution of my PhD consists in importing

logical methods to bypass the need for stabilization and prove the complete unsoundness of S and $\mathscr{R}$. System $\mathscr{R}$ is actually unfit to formulate the proof of complete unsoundness and, as for Klop's Problem, it seems unavoidable to work with System S.

The Collapse of S onto $\mathscr{R}$

If we forget about tracks, a sequence collapse on a multiset *e.g.* $(2 \cdot S, 3 \cdot T, 8 \cdot S)$ collapses on $[S, T, S]$. If this collapse is performed coinductively, then any derivation of S will collapse on a derivation of $\mathscr{R}$. We may ask if the converse is true: is any derivation of $\mathscr{R}$ the collapse of some S-derivation ? That is: is the collapse of System S onto System $\mathscr{R}$ surjective ? The answer is positive:

Theorem (Vial)**.** Every $\mathscr{R}$-derivation is the collapse of a S-derivation and we can encode in system S any dynamical behaviour of System $\mathscr{R}$.

This is also a problem in which no form of stabilization is ensured and we must also resort to logical methods to prove this theorem. This work also features an auxiliary type system (a variant of System S) that encodes "reduction choices" that exist with multiset intersection.

Resource types for classical natural deduction (joint work with Delia Kesner)

With Delia Kesner, we designed quantitative type system $\mathcal{H}_{\lambda_\mu}$ and $\mathcal{S}_{\lambda_\mu}$ for λ_μ-calculus, featuring intersection and union types and extending system $\mathscr{R}_0$, such that:

Theorem (Kesner,Vial, FSCD17)**.**

- A λ_μ-term t is head normalizing iff it is typable in $\mathcal{H}_{\lambda_\mu}$.
- A λ_μ-term t is strongly normalizing iff it is typable in $\mathcal{S}_{\lambda_\mu}$.

As in other non-idempotent frameworks, those type systems provide upper bounds for the length of some normalizing reduction sequences (*e.g.* the head reduction sequence for head normalizing terms) and relies on simple arithmetical arguments to prove termination. Along with those type systems, we also developed an explicit substitution and replacement classical calculus that extends λ_μ, as well as extensions of $\mathcal{H}_{\lambda_\mu}$ and $\mathcal{S}_{\lambda_\mu}$ with the expected type characterizations.

Polyadic Approximations, Fibrations and Intersection Types (joint work with Damiano Mazza and Luc Pellissier)

Starting from an exact correspondence between affine approximations and non-idempotent intersection types, we have developed a general framework for building intersection types systems characterizing normalization properties in the finite case. This construction, which uses in a fundamental way Mellis and Zeilbergers "type systems as functors" viewpoint, allows us to recover equivalent versions of every well known intersection type system (including Coppo and Dezanis original system, as well as its non-idempotent variants independently introduced by Gardner and de Carvalho).

An intersection type system is then defined as a colored operads (*i.e.* a symmetric multicategory with possibly more than one object) endowed endowed with 2-arrows, that is, arrows between arrows: a multimorphisms generalizes the notion of typed judgment $\Gamma \vdash t : B$, where t is now thought as a multimorphisms from Γ to B. Intuitively, there is a 2-arrow from $\Gamma \vdash t : B$ to $\Gamma \vdash t' : B$ if the latter judgment is obtained from the former by subject reduction steps.

An intersection type system is built from a given restriction of the λ-calculus (*e.g.* the set of λ-terms endowed with head reduction) and a given subset of the simply typed polyadic calculus by a pull-back construction. This framework enables to almost automatically build new systems of intersection types in this way. The proofs of the termination properties (*i.e.* properties of the form "if t is typable, then t is normalizing") in most known type systems are now subsumed by the proof of strong normalization of the propositional multiplicative exponential linear logic.